This is an Author's Accepted Manuscript of an article published in:

Cybernetics and Systems: An International Journal, 2011, vol. 42 (8), 636-658. ISSN: 1087-6553.

DOI: http://dx.doi.org/10.1080/01969722.2011.634681

This work has been supported by the CAM Project S2009/DPI-1559/ROBOCITY2030 II, and "A New Approach to Social Robotics" (AROS), of MICINN (Ministry of Science and Innovation), developed by the research team RoboticsLab at the University Carlos III of Madrid.

Learning to avoid risky actions

María Malfaz and Miguel A. Salichs

RoboticsLab, Carlos III University of Madrid
28911, Leganés, Madrid, Spain
mmalfaz@ing.uc3m.es ; salichs@ing.uc3m.es

Abstract

When a reinforcement learning agent executes actions that can cause frequently damages to itself, it can learn, by using Q-learning, that these actions must not be executed again. However, there are other actions that do not cause damage frecuently, only once in a while: risky actions, such as parachuting. These actions may imply a big punishment to the agent and, depending on its personality, it would be better to avoid. Nevertheless, using the standard Q-learning algorithm the agent is not able to learn to avoid them, since the result of these actions can be positive in average. In this paper, an additional mechanism to Q-learning, inspired by the emotion of fear, is introduced in order to deal with those risky actions by considering the worst results of them. Moreover, there is a daring factor for adjusting the consideration of the risk. This mechanism is implemented on an autonomous agent living in a virtual environment. The results present the performance of the agent with different daring degrees.

1 Introduction

In this research, it is considered that a risky action is one that, in average, can produce positive results but, once in a while, generates very bad results for the agent.

Let us show an example of an action that can become risky. It can happen that one likes mushrooms: they taste good and satisfy hunger. But if on some occasion one suffers an intoxication due to these mushrooms, one will not probably eat them again, although that happened just one time. Therefore, the action of eating mushrooms becomes risky.

When an action executed by the agent has bad results frequently, the agent, by using Q-learning [Watkins, 1989], learns not to select it. This is due to the very low value of the action as a consequence of the negative reinforcement received when it is executed. A different situation occurs when the action is risky. In this case, the agent can learn that the long term value of this action is high, although it receives very negative reinforcements with a small probability. This is because this learning algorithm does not take into account the worst experiences when the agent executed this kind of action, just the average value.

In this paper, we present a model-free method in order to find out the optimal policy in the described situation. A model-free method implies that the agent knows neither the state transition probability function nor the reinforcement function [Kaelbling et al., 1996]. We have developed an additional mechanism to the standard Q-learning algorithm in order to consider not only the average value of the actions but also their worst results. In this mechanism, a daring factor β will adjust the importance that the agent gives to both contributions.

In order to test this mechanism, it is implemented in an autonomous agent living in a virtual environment. In this approach, it is considered that the agent is motivationally autonomous. According to Gadanho [Gadanho, 1999] and Cañamero [Cañamero, 2003], an autonomous agent has goals and motivations and it has some ways to evaluate its behaviours in terms of the environment and its own motivations. Its motivations are desires or preferences that can lead to the generation and adoption of objectives. The final goals of the agent, or its motivations, must be oriented to maintain its internal equilibrium.

In this research, the agent lives in a virtual world surrounded by objects that it needs in order to satisfy its necessities (e.g., food for satisfying hunger). The agent knows the properties of every object, that is, it knows which actions can be executed with each object (affordances). What the agent does not know is which action is more appropriate in each situation. It must learn a policy of behaviour in order to maintain all its needs within acceptable ranges. The policy establishes a normative about what to do in each situation. This means that the agent must learn the proper relation between states and actions by evaluating the long term value of executing an action in a certain state for every action-state pair.

In this paper, the performance of an autonomous agent considering the worst effects of its actions is studied. Since the value of the daring factor β is an important parameter for the final design of the agent, its performance using several values of this factor is presented.

The paper is organized as follows. First, section 2 introduces the proposed mechanism to complete the Q-learning algorithm in order to deal with risky ac-

tions. Next, in section 3, a brief review of some related works is given. Section 4 shows the experimental procedure used in this work. First, the environment, the virtual world where the agent lives, is brief y described. Then, a general view of how the agent is modeled as well as a description of its learning process are given. Moreover, the adaptation of the proposed mechanism to the current application is presented. The results obtained varying the daring factor of the agent are presented in section 5. Finally, the main conclusions of this paper are summarized in section 6.

2 The proposed method to deal with risky actions

As previously stated, the agent learns how to behave in every possible situation by using the Q-learning algorithm [Watkins, 1989]. The goal of this algorithm is to estimate the $Q(s, a)$ values. These values are the expected reward for executing the action a in the state s and then following the optimal policy from there. Every $Q(s, a)$ value is updated according to

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma V(s')) \tag{1}$$

where

$$V(s') = \max_{a \in A} \big(Q(s', a)\big) \tag{2}$$

is the value of the new state s' and is the best reward the agent can expect from this state. A is the set of actions, r is the reinforcement, γ is the discount factor, and α is the learning rate.

Once the optimal function Q, is obtained, then it is easy to calculate the optimal policy, by observing every possible action in a certain state and selecting the one that:

$$maximizes\,(Q(s, a)) \tag{3}$$

The proposed method is inspired in the emotion of fear. When one is afraid of executing an action, it is because, one knows that this action can have a noxious effect. This effect constitutes a negative reinforcement. As already stated, by using Q-learning, the agent learns the average values of its actions. Therefore, if an action is frequently noxious, the agent learns to avoid it. But when the action causes negative effects only occasionally, then another mechanism is needed in order to learn to deal with those risks. The agent must learn what to do with these risky actions.

It is important to note that these kind of actions are executed by the agent and the next state reached by it is a consequence of its action. Therefore, this must not be confused with rare events, since according to Frank et al [Frank et al., 2008], those events occur independently from the actions of the agent.

The idea of these risky actions is that, in general, they have a positive effect, but once in a while they cause a very negative reward. Therefore, in order to consider these very negative effects on the agent, the worst results experimented by it, and for every state-action pair, are stored in a variable called $Q_{worst}(s, a)$. This variable is updated after the execution of the action.

$$Q_{worst}(s,a) = \min(Q_{worst}(s,a), r + \gamma \cdot V_{worst}(s')) \tag{4}$$

where

$$V_{worst}(s') = \max_{a \in A}(Q_{worst}(s',a)) \tag{5}$$

is the worst value of the new state s' and it is considered as the best the agent can do with the worst Q values.

Let us def ne a new Q value, $Q_{fear}(s,a)$, that includes the average Q value, calculated using Q-learning by (1) and (2), as well as the worst Q value, calculated by (4) and (5):

$$Q_{fear}(s,a) = \begin{cases} \beta \cdot Q(s,a) + (1-\beta) \cdot Q_{worst}(s,a) & If\ Q_{worst}(s,a) \leq L_m \\ \\ Q(s,a) & In\ other\ cases \end{cases} \tag{6}$$

where L_m is a user specif ed threshold. According to (6), only when the worst value of the action a executed in the state s is lower than the limit L_m, then the agent takes into account the worst Q value as well as the optimal Q value.

In this research, in order to deal with risky and non-risky actions, the agent chooses the action that:

$$maximizes\,(Q_{fear}(s,a)) \tag{7}$$

Parameter β, denominated the daring factor, being $0 \leq \beta \leq 1$, measures the daring degree of the agent. It can be noticed that, when $\beta = 1$, the agent uses the Q values calculated by the standard Q-learning. In fact, if β is near 1, the agent is very daring, or risk-seeking, since it barely takes into account the worst result of the action; on the contrary, for a risk-adverse agent that tries to minimize the risk, β will be near 0.

This daring factor depends, in the case of humans, on the personality. In real life, there are some activities that could be considered as risky, such as climbing, parachuting, bungee jumping, and so on. Nevertheless, there are many people who like those activities very much without considering the possible risks and their fatal consequences that may lead to death. On the other hand, many other people consider that the rush of adrenaline caused by those activities do not compensate the risk.

3 Related work

The algorithm proposed in this paper can be seen as a new approach to risk-adverse reinforcement learning. One of the most popular proposals was presented by Heger [Heger, 1994]. He proposed a new algorithm called *$\hat{Q}$-learning* to f nd out the policy that minimizes the worst-case total discounted costs. The $\hat{Q}$ value is updated as follows:

$$\hat{Q}(s,a) = \max\left(\hat{Q}(s,a), c + \gamma \cdot \min_{b \in A} \hat{Q}(s,b)\right) \tag{8}$$

where c is the immediate cost. The objective is to learn optimal actions so as to

$$minimize\left(\max\left(c+\gamma\cdot\min_{b\in A}\hat{Q}(s,b)\right)\right) \tag{9}$$

As can be easily observed, our Q_{worst} value is calculated by using (4) and (5) in a very similar way to this $\hat{Q}$ value. In fact, both algorithms are the same when the daring factor β is equal to 0. Note that in our case, we try to maximize the minimum reward, while Heger tries to minimize the maximum cost. Moreover, the algorithm proposed by Heger only cares about avoiding the risk, while the one proposed in this paper offers the possibility of selecting how the risk affects the decision making process. Depending on the value of the daring factor, the agent can be risk-adverse or not. These kind of algorithms are denominated *risk-sensitive*.

There are several risk-sensitive approaches, such as the ones introduced by Howard and Matheson [Howard and Matheson, 1972] and Coraluppi and Marcus [Coraluppi and Marcus, 1999]. Both proposals make use of *exponential utility functions*. The idea is to transform the cumulative returns by exponential utility functions and seek optimal policies with respect to this utility measure. The risk-sensitive objective is given by:

$$minimize\frac{1}{\delta}\log E\left[e^{\delta\cdot\sum_k c_k}\right] \tag{10}$$

Again, c represents the immediate cost. Besides, there is a risk-aversion coeff cient δ. When it is small, the value to be minimized in (10) takes the form:

$$E\left[\sum_k c_k\right]+\frac{\delta}{2}Var\left[\sum_k c_k\right] \tag{11}$$

Therefore, for $\delta > 0$ the variability in the cost is penalized, so the agent is risk-adverse. The *expected value-minus-variance-criterion* proposed by Heger has a similar effect to the expected-utility one:

$$E(R)-kVar(R) \tag{12}$$

where k is called the risk-adverse factor and R is the reward, see [Heger, 1994].

In these cases, the agent needs the transition and reinforcement probabilities of the environment, in contrast with our approach, that is model free.

Mihatsch and Neuneier [Mihatsch and Neuneier, 2002] also proposed a risk-sensitive reinforcement learning algorithm based on a very different philosophy. Instead of transforming the cumulative return of the process as in the utility theory, they *transform the temporal differences* (so-called TD-errors) which play an important role during the procedure of learning the value. Moreover, they are able to formulate risk-sensitive versions of Q-learning and TD-learning. The transformation function is def ned as follows:

$$\chi^k : x \mapsto \begin{cases} (1-k)x & if\ x>0 \\ (1+k)x & otherwise \end{cases} \tag{13}$$

where $k\,\epsilon\,(-1,1)$.

The risk-sensitive Q-learning algorithm proposed updates the Q values as follows:

$$Q(s,a) = Q(s,a) + \alpha \cdot \chi^k \left(r + \gamma \max_a Q(s',a) - Q(s,a) \right) \quad (14)$$

The only difference from traditional Q-learning is in the transformation χ^k that weights positive and negative temporal differences appropriately. Setting $k = 0$, we recover the original Q-learning algorithm. If k is set to be positive, then the negative temporal differences are overweigh with respect to positive ones. Loosely speaking, they overweigh transitions to successor states where the immediate return r happened to be smaller than in the average. On the other hand, they underweigh transitions to states that promise a higher return than in the average. In other words, the objective function is risk-avoiding if $k > 0$ and risk-seeking if $k < 0$.

Again, the optimal action is the one that, for every state:

$$maximizes\,(Q(s,a)) \quad (15)$$

In this case, the main difference with respect to our algorithm is that after the learning process is f nished, the learnt Q values are dependent on the selected value of k. Using our approach, the optimal Q values and the Q_{worst} values are calculated. Therefore, if we want to calculate the Q_{fear} values using a different risk-sensitive factor or daring factor β, we do not need to launch the learning process again. Moreover, our proposal allows the possibility of varying the daring factor during the life of the agent.

On the other hand, there is another different approach by Geibel [Geibel, 2001] [Geibel and Wysotzki, 2005] that def nes the risk as the probability of entering a fatal state. They consider the problem of f nding out optimal policies with *bounded risk*, that is, the risk is smaller than some user-specif ed threshold ω. Finally, they propose a learning algorithm in such a way that if the agent is in a state s, then it selects an action a that

$$maximizes\,(\lambda Q(s,a) - \sigma(s,a)) \quad (16)$$

where $Q(s,a)$ are the values obtained using the Q-learning algorithm and $\sigma(s,a)$ is the probability that the agent ends in a fatal state. Parameter λ can be used to increase or decrease the inf uence of the Q values compared to the σ values.

The idea is to start the learning process with $\lambda = 0$ and to increase its value until the number of ω-safe states for the current policy decreases. This proposal is very interesting, although, as said before, the risk is related to a very different concept: entering a pre-def ned set of fatal states. In our approach, as in many others, the risk is related to the execution of certain actions that may lead to bad rewards. Moreover, there is no pre-def ned information about the states.

4 Experimental test-bed

As previously said, the proposed mechanism for dealing with risky actions is tested on virtual autonomous agents who are living in a virtual environment. This experimental platform has been already used to test a decision making system for

an autonomous agent based on drives, motivations and emotions [Malfaz and Salichs, 2006] [Malfaz and Salichs, 2010][Salichs and Malfaz, 2011]. The main difference of the presented work with the previously refereed ones is that in this new implementation, some risky actions are included, so the proposed mechanism is needed. In this section, a brief description of both, the virtual environment and the description of the drives and motivations of the agents, are presented. Moreover, the implementation of the proposed additional mechanism to the learning process is also described.

4.1 The virtual environment

In order to create the environment, we use a role playing game based on text, available on the net and called CoffeMud [Zimmerman, 2007]. In this environment, formed by corridors and rooms, see f gure 1, the player can f nd different objects: food, water, medicine, elixir, and world.

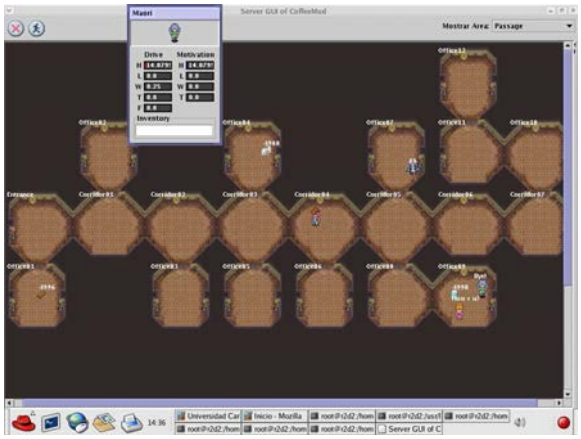

Figure 1: Virtual environment

The food, the water, the medicine, and the elixir are distributed in rooms in such a way that there is a room with food, another with medicine, another with water, and f nally another with elixir. The amount of objects present in those rooms is huge and therefore, it is considered that the agent has unlimited resources. The agent, at the beginning of its life, does not know where to f nd those objects. Throughout its life time, it f nds the objects and remembers their position. Therefore, if the agent needs an object, it will know where to f nd it. This virtual world is grid-based, as can be observed in f gure 1, and the agent moves around by sending ‘north’, ‘south’, east’, and ’west’ commands.

4.2 Drives and motivations of the agent

As stated in section 1, it is considered that an autonomous agent has certain needs (drives) and motivations. The goal of the agent will be to learn to select the right action in every state in order to maintain those needs within an acceptable range.

4.2.1 Drives

The considered drives and motivations are the following:

- Hunger
- Thirst
- Weakness

The Hunger and the Thirst drives are obviously related to the lack of resources and their values increase as the agent spends time without consuming food or water. The Weakness drive, on the other hand, is related to the need of recovery of the agent.

The values of the Hunger and the Thirst drives increase a certain amount at every step simulation. These drives do not grow at the same rate. Physiological studies determine that in most human beings the necessity of water (thirst) appears before the necessity of food (hunger). In [Gautier and Boeree, 2005], it is presented how Maslow discovered that certain needs prevail over others. For example, if one is hungry and thirsty, one will tend to relieve thirst before hunger. As a conclusion, the Thirst is a stronger drive than the Hunger.

On the other hand, these drives, after being satisf ed (their values become zero), do not start to increase their values immediately but after a certain time, which we call "satisfaction time". This happens in the same way as after eating, since one is not hungry again until some hours later. In the next equation, the satisfaction times corresponding to these drives are shown:

$$\begin{aligned} T_{thirst} &= 50\ steps \\ T_{hunger} &= 100\ steps \end{aligned} \tag{17}$$

According to these values, the Thirst drive is the most urgent one, since it takes less time to increase its value again. As already explained, one is thirsty more frequently than hungry. Once the satisfaction time passes, the drives grow as follows:

$$\begin{aligned} D_{thirst}^{k+1} &= D_{thirst}^{k} + 0.1 \\ D_{hunger}^{k+1} &= D_{hunger}^{k} + 0.08 \end{aligned} \tag{18}$$

As shown, the growing rate of the Thirst drive is higher than that of the Hunger drive.

On the other hand, the variation of the Weakness drive depends on the movement of the agent. Therefore, if the agent stands still, this drive does not suffer any variation, but if the agent moves, the value of the drive increases at every step, as shown next:

$$D_{weakness}^{k+1} = D_{weakness}^{k} + 0.05 \tag{19}$$

As shown, the growing rate of the Weakness drive is lower than the ones of the Hunger and Thirst drives. Moreover, this drive does not need satisfaction time.

4.2.2 Motivations of the agent

The motivational states represent tendencies to behave in particular ways as a consequence of internal (drives) and external (incentive stimuli) factors [Ávila García and Cañamero, 2004]. In other words, the motivational state is a tendency to correct the error, that is, the drive, through the execution of behaviours.

In this approach, in order to model the motivations of the agent, we use Lorentz's hydraulic model of motivation as an inspiration [Lorenz and Leyhausen, 1973]. Therefore:

$$\begin{aligned} &\text{If } D_i < L_d \text{ then } M_i = 0 \\ &\text{If } D_i \geq L_d \text{ then } M_i = D_i + w_i \end{aligned} \tag{20}$$

where M_i are the motivations, D_i are the related drives, w_i are the related external stimuli, and L_d is called the activation level.

These external or motivational stimuli, w_i, are the different objects that the agent can f nd in the world during its life, so:

$$\begin{aligned} &\text{If the stimuli are present, then } w_i = 1 \\ &\text{If the stimuli are not present, then } w_i = 0 \end{aligned} \tag{21}$$

Table 1 shows the motivations/drives and their related motivational stimuli.

Table 1: Motivations, Drives, and Motivational stimuli

Motivation/Drive	Motivational stimuli
Hunger	Food & elixir
Thirst	Water & elixir
Weakness	Medicine & elixir

The activation level L_d, used in (20) for calculating the value of the intensity of motivations, is set as follows:

$$L_d = 2 \tag{22}$$

These values of the external stimuli and the activation level have been selected based on previous experiments where several values were tested.

4.3 Implementation of the learning algorithm with the additional mechanism for dealing with risky actions

4.3.1 Reinforcement function

In this research, the wellbeing of the agent is def ned as the degree of needs satisfaction and is calculated as follows:

$$Wb = Wb_{ideal} - (\alpha_1 D_{hunger} + \alpha_2 D_{thirst} + \alpha_3 D_{weakness}) \tag{23}$$

where $Wb_{ideal} = 100$ is the ideal value of the wellbeing of the agent and, α_i are the ponder factors that measure the importance of each drive on the wellbeing of the

agent. In the experiments, all the drives will have the same importance. Therefore, all the ponder factors are equal to one another:

$$\alpha_i = 1 \tag{24}$$

The wellbeing is def ned this way since, logically, as the needs of the agent increase, its wellbeing must decrease. Therefore, when all the drives of the agent are satisf ed, their values are zero and the wellbeing is at its maximum. It can happen that the value of the wellbeing of the agent is negative.

Since the goal of the agent is to learn behave in order to satisfy its needs, it can be also said that the f nal goal is to learn to select the right action in every state in order to maximize the wellbeing of the agent. Therefore, as a f rst idea, it seemed logical to think of using the wellbeing of the agent as the reinforcement function, since it gives information about the effect of an executed action on the agent. In fact, Gadanho [Gadanho and Custodio, 2002] def nes a wellbeing signal generated in a similar way and it was used as the reinforcement function in a reinforcement learning frame. Nevertheless, for our research, it seems to be more appropriate to use the variation of the wellbeing as the reinforcement function. In fact, this variation gives a clearer idea about how an action affects the wellbeing of the agent.

This variation of the wellebing ($\triangle Wb$) is calculated as the current value of the wellbeing minus its value in the previous step, as shown in the next equation:

$$\Delta Wb^{k+1} = Wb^{k+1} - Wb^{k} \tag{25}$$

The biggest positive variation of the wellbeing will be produced when the drive with the highest intensity is satisf ed.

4.3.2 State of the agent

In this system, the state of the agent is def ned as the combination of its inner state, S_{inner}, and its external state, $S_{external}$, as shown in (26). The inner state of the agent is related to its internal needs (e.g., the agent is hungry) and the external state is the state of the agent in relation to all the objects present in the environment (e.g., the agent has food and water):

$$S = S_{inner} \times S_{external} \tag{26}$$

The inner state is determined by the motivation with the highest intensity, that is, the dominant motivation, as shown in (27). According to (20), if none of the drives is high enough, then all $M_i = 0$, and there is not any dominant motivation. In this case it can be considered that the agent has no needs, it is "OK".

$$S_{inner} = \begin{cases} \arg\max_i M_i & If\ \max_i M_i \neq 0 \\ OK & In\ other\ cases \end{cases} \tag{27}$$

According to the considered motivations, the inner state of the agent is def ned as follows:

$$S_{inner} = \{Hungry,\ Thirsty,\ Weak,\ OK\} \tag{28}$$

On the other hand, the external state, as we have just said, is the state of the agent in relation to all the objects:

$$S_{external} = S_{obj_1} \times S_{obj_2} ... \tag{29}$$

In previous works [Malfaz and Salichs, 2006] [Malfaz and Salichs, 2009], it was explained that, in order to reduce the complexity of the learning process, the states related to the objects are considered as independent from one another. Therefore, the agent learns how to behave in relation to every object separately. Then, for example, the agent learns what to do with food independently from learning what to do with water. This implies that the Q values are now calculated for each object in every inner state. From now on, the nomenclature for the Q values will be $Q^{obj_i}(s, a)$. The super-index obj_i specifies the object that the agent is dealing with, $a \in A_{obj_i}$, where A_{obj_i} is the set of actions related to the object i, and s is the total state of the agent in relation to object i and is defined as follows:

$$s \in S_{inner} \times S_{obj_i} \quad \forall i \tag{30}$$

Therefore, the agent will learn, for example, what to do with food when it is hungry, with food when it is thirsty, with water when it is hungry, and so on.

The state related to every object, except for the *world* object, is the combination of three binary variables:

$$S_{obj} = Being\_in\_posession\_of \times Being\_next\_to \times Knowing\_where\_to\_find \tag{31}$$

The state of the agent in relation to the *world* object is unique: the agent is always in the world.

$$S_{world} = Being\_at \qquad (Always\ True) \tag{32}$$

4.3.3 Actions of the agent

The sets of actions that the agent can execute, depending on its state in relation to the objects, are the following:

$$A_{food} = \{Eat,\ Get,\ Go\ to\} \tag{33}$$

$$A_{water} = \{Drink\ water,\ Get,\ Go\ to\} \tag{34}$$

$$A_{medicine} = \{Drink\ medicine,\ Get,\ Go\ to\} \tag{35}$$

$$A_{elixir} = \{Drink\ elixir,\ Get,\ Go\ to\} \tag{36}$$

$$A_{world} = \{Stand\ still,\ Explore\} \tag{37}$$

The "explore" and "go to *name of the room*" actions are sequences of movement commands. The "explore" action tries to reach every room of the environment and the "go to" action gives the sequence of movement commands to reach the target room following the shortest path.

Among all these behaviours, there are some of them that cause an increase or decrease of some drives, as shown in table 2, leading to a variation in the wellbeing of the agent.

Table 2: Effects of the actions on drives

Action	Drive	Effect
Eat	Hunger	Reduce to zero (drive satisfaction)
Drink water	Thirst	Reduce to zero (drive satisfaction)
Drink medicine	Weakness	Reduce to zero (drive satisfaction)
Drink elixir	Hunger Thirsty Weakness	95% of times: Reduce to zero (drive satisfaction) 5% of times: Increase 4 units
Explore/ go to	Weakness	Increase 0.05 per each step

As can be observed, the actions related to the consumption of food, water, and medicine always satisfy drives and so, they will produce a positive reward. On the other hand, the consumption of the elixir produces, most times, a big positive effect on the wellbeing of the agent, satisfying Hunger, Thirst, and Weakness. Nevertheless, the elixir occasionally works as a "poison" that leads to an increment of Hunger, Thirst, and Weakness. Those increments of 4 units, compared with the amounts increased at every simulation step, see (18) and (19), are very high. This implies that the wellbeing of the agent suffers a decrement of 12 units of magnitude and so, it is a big punishment.

Therefore, according to the definition given in section 1 and considering the effects of the actions shown in table 2, we can conclude that there is only a risky action: to drink elixir. The rest of actions have always the same effect, although in the case of the ones related to the consumption of objects, we cannot say that they have the same positive reward, since it will depend on the current value of the satisfied drive.

4.3.4 Learning to take into account the risky actions

Considering the assumptions made in relation to the external state, the proposed method for dealing with risky actions is modified. First, the worst results

experimented by the agent during its life, for every state-action pair, are stored in the next variable:

$$Q^{obj_i}_{worst}(s_{obj_i}, a) = \min(Q^{obj_i}_{worst}(s_{obj_i}, a), r + \gamma \cdot V^{obj_i}_{worst}(s'_{obj_i})) \tag{38}$$

where

$$V^{obj_i}_{worst}(s'_{obj_i}) = \max_{a \in A_{obj_i}} (Q^{obj_i}_{worst}(s'_{obj_i}, a)) \tag{39}$$

is the worst value of object i in the new state s' and it is considered as the best the agent can do with the worst Q values. This worst value is calculated for every object separately.

These values are calculated for every object of the environment, not for every inner state and object. This is because they do not depend on the inner state of the agent. The idea is the following: if one is hungry and eats a mushroom that causes him a stomachache, one will try to avoid mushrooms always, not just when hungry.

Now, the proposed method inspired by the emotion of fear def ned by (6) is also modif ed as follows:

$$Q^{obj_i}_{fear}(s, a) = \begin{cases} \beta \cdot Q^{obj_i}(s, a) + (1 - \beta) \cdot Q^{obj_i}_{worst}(s_{obj_i}, a) & If\ Q^{obj_i}_{worst}(s_{obj_i}, a) \leq L_m \\ Q^{obj_i}(s, a) & In\ \ other\ \ cases \end{cases} \tag{40}$$

where L_m is the same threshold introduced in (6). According to (40), only when the worst effect of the action a related to object i and executed in the state s_{obj_i} is lower than the limit L_m, then the agent takes into account the worst Q value as well as the average Q value. It must be said that, in our experiments, there is not any action that always causes negative affects. Therefore, we have two kinds of actions: those that always cause positive effects (non-risky actions) and those that, in average, cause positive effects but occasionally produce very negative rewards (risky actions). As a consequence, when the worst effect of an action is lower than L_m, it will be considered that this action is a risky one.

As stated in section 2, the agent chooses the action that maximizes $Q^{obj_i}_{fear}(s, a)$. Using this approach, if the action is considered as risky, the expected result of the action is considered at the same time as the least favorable one.

5 Experimental results

In this section, the procedure followed in this experiment is shown. The life of the agent consists of two phases: the learning phase and the steady phase.

- During the learning phase, the agent, by using Q-learning, learns the long term value of every action in every state $Q^{obj_i}(s, a)$ by exploring all the state-action pairs. The agent starts with all the initial Q values equal to zero. Through its experience in the world, it learns and updates its Q values. It tries out actions probabilistically based on the Q values using a Boltzmann distribution [Watkins, 1989]. In this distribution, there is a parameter called

temperature that can be tuned. At the beginning of this phase the temperature is high in order to favor the exploration of every action. Along this phase this temperature decreases gradually for the exploitation of the most suitable actions. Moreover, the value of the learning rate α also decreases gradually from the value $0,3$ till 0. The value of the discount factor γ is set to $0,8$ during the entire phase.

At the same time, in order to implement the mechanism described in section 4.3.4, the agent stores the worst experienced results for each state-action pair $Q^{obj_i}_{worst}(s_{obj_i}, a)$.

- During the steady phase, the agent lives using the values learnt in the learning phase by selecting the actions that maximize the $Q^{obj_i}_{fear}(s, a)$ def ned in (40). It is important to note that the agent will select among all the available actions related to every object. Therefore, at one time the agent can interact with food and in the next step with water, for instance. It will select the action with the highest $Q^{obj_i}_{fear}(s, a)$ value. In relation to the learning parameters, the value of the learning rate α is set to 0 and the discount factor γ is set to $0,8$.

  During this phase, the daring factor β varies in order to observe the performance of the agent with different degrees of courage. This daring factor, def ned in section 4.3.4, ponders the worst value of the action, $Q^{obj_i}_{worst}(s_{obj_i}, a)$, and its average value, $Q^{obj_i}(s, a)$.

The results presented in this section corresponds to one trial since it is better to analyze the performance of the agent in more detail, in order to understand the algorithm proposed.

In this experiment, the limit L_m used in (40) has been f xed, after several experiments, to a value equal to the negative reinforcement that the agent receives when the elixir works as a poison: $L_m = -12$

In f gure 2, the evolution of the wellbeing of the agent during both phases is shown. The learning phase goes from the beginning of its life till 45000 simulation steps and the steady phase from 45000 till 75000.

As can be observed, during the learning phase the wellbeing of the agent presents important drops. These decrements are mostly due to the times that the agent drank the elixir when it was a poison, causing the picks in the Hunger, Weakness, and Thirst drives, see f gure 3. It must be said that, although in this experiment the agent cannot die, the negative reinforcement received due to the poisoning is signif cantly high. Therefore, in order to analyse the results, we must consider the number of poisonings during the steady phase.

In this experiment, the steady phase has been divided into different zones with different values of the daring factor β. As shown in f gure 4, during the steady phase the wellbeing varies as the daring factor decreases. This variation has a forward relationship with the number of times that the agent drank the elixir and disagreed with it. In fact, it can be observed that as the daring factor decreases, the drops in the wellbeing due to the poisoning disappear.

In table 3, the values of the daring factor β and the number of times that the agent drank the elixir are shown, as well as the number of poisonings (number of drops). The shown values conf rm that as the daring factor decreases, the agent

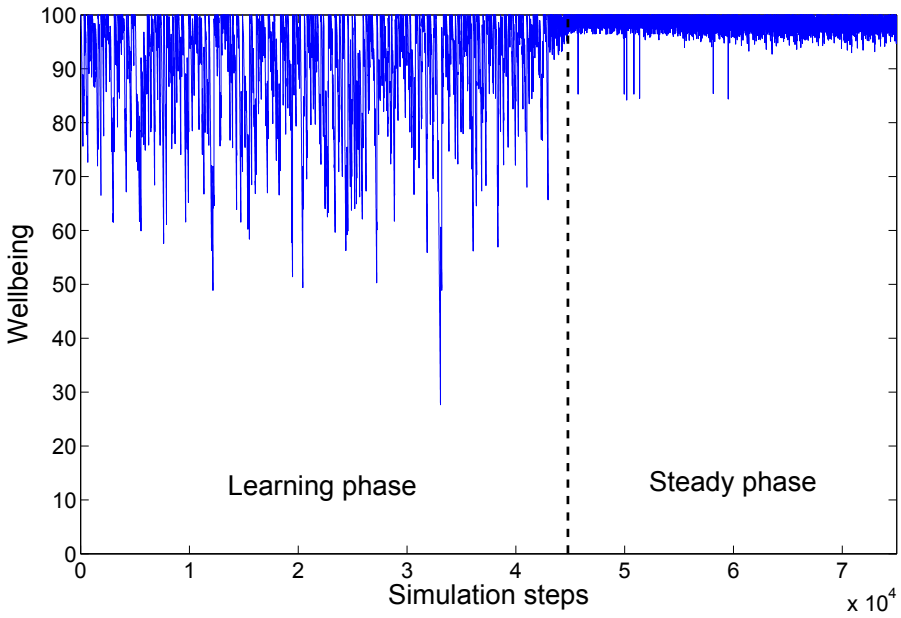

Figure 2: Wellbeing of the agent

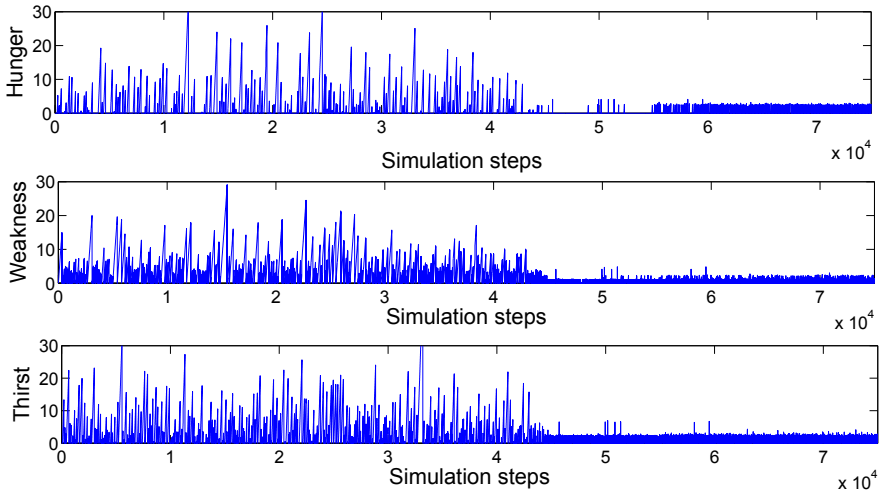

Figure 3: Drives of the agent

stops drinking the elixir, since it becomes risk-adverse. This means that it is more concerned about the possibility of being poisoned and the number of drops in the wellbeing disappears.

Let us analyze some $Q_{fear}^{obj_i}(s, a)$ values in order to understand the results better. First, the worst $Q_{worst}^{obj_i}(s_{obj_i}, a)$ values of the actions related to the elixir are shown in f gure 5. As was expected, these values are all lower than the limit introduced in (5). On the other hand, the worst registered values for the actions related to food, water, and medicine are higher than the limit L_m given by (5). This is because nothing bad happens when the agent executes actions with those objects.

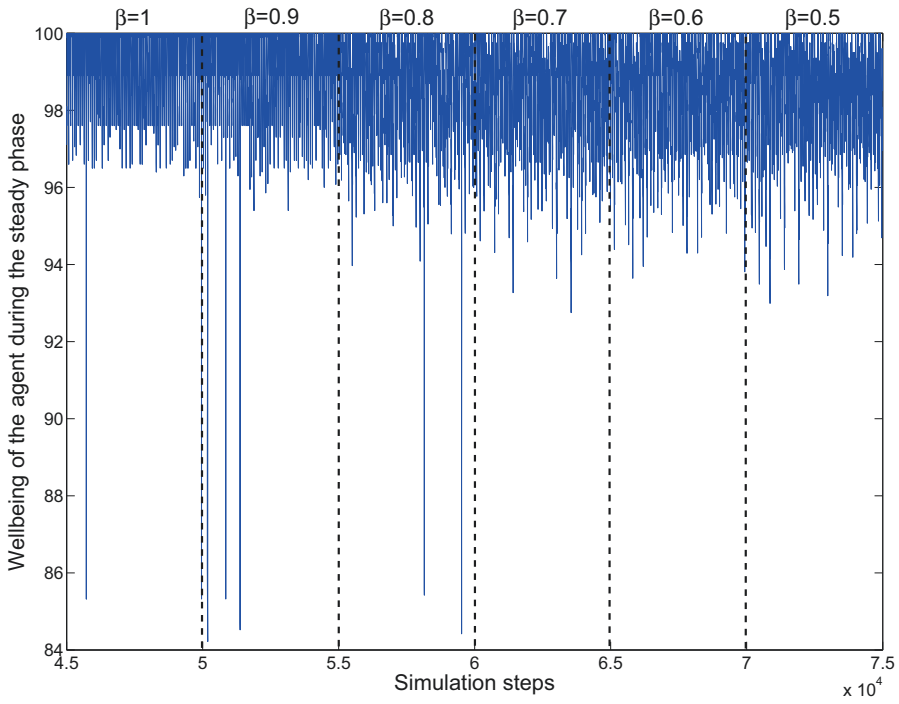

Figure 4: Wellbeing of the agent during the steady phase as the daring factor β varies

Table 3: Results of the steady phase

Steady phase	*Value of* β	*N^o of times that the agent drank the elixir*	*N^o of poisonings*
from 45000 to 50000	1	68	2
from 50000 to 55000	0.9	68	3
from 55000 to 60000	0.8	21	2
from 60000 to 65000	0.7	0	0
from 65000 to 70000	0.6	0	0
from 70000 to 75000	0.5	0	0

Figure 6 shows the $Q^{obj_i}(s, a)$ values of the actions related to the elixir when the dominant motivation is Hunger. As can be observed, those values are very high. This is because, as previously explained, the learnt values are the average value of the reinforcements received. In general, as observed in table 3, the number of poisonings is very small in comparison with the number of times that the elixir satisf ed all the needs of the agent. Therefore, in average, the effects of the elixir are very positive.

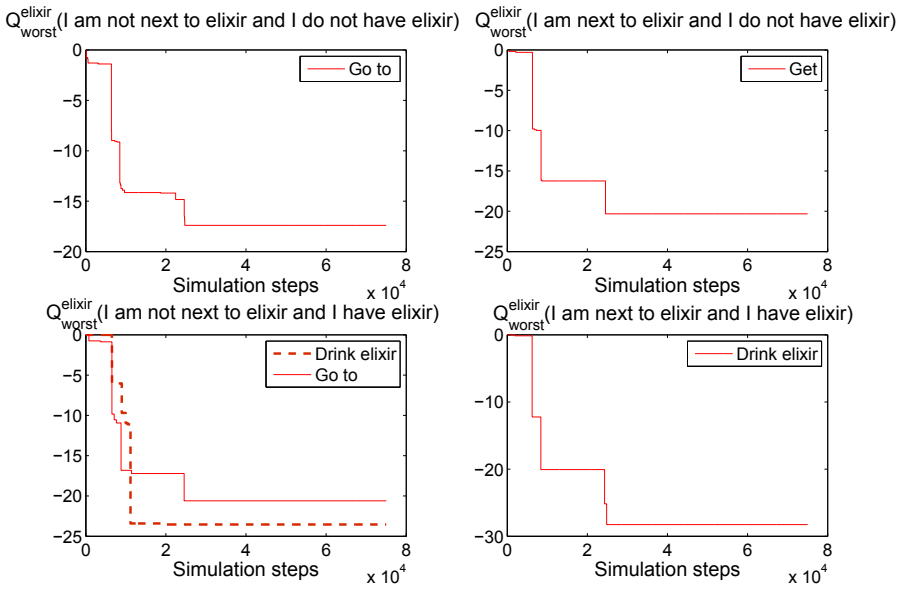

Figure 5: The $Q^{obj_i}_{worst}(s_{obj_i}, a)$ values of the actions related to the elixir

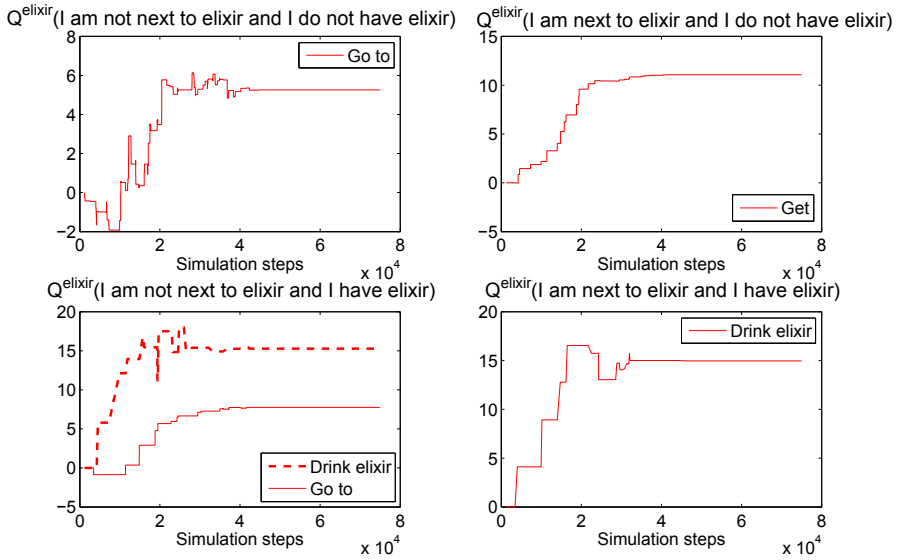

Figure 6: $Q^{obj_i}(s, a)$ values of the actions related to the elixir when the dominant motivation is the Hunger

Next, the analysis of the $Q^{obj_i}_{fear}(s, a)$ values of the actions related to the food and the elixir when the agent is hungry is going to be presented. Figure 7 shows the $Q^{obj_i}_{fear}(s, a)$ values of the actions related to the food (left part of the graph) and the elixir (right part of the graph). In every sub-graph, the steady phase is divided into different zones with the previously selected values of the daring factor β. As shown, the $Q^{obj_i}_{fear}(s, a)$ values of the actions related to the food do not change as the daring factor decreases, since the $Q^{obj_i}_{worst}(s_{obj_i}, a)$ values are higher than the limit

introduced in (5). As a consequence, according to (40): $Q_{fear}^{obj_i}(s,a) = Q^{obj_i}(s,a)$.

On the contrary, the $Q_{fear}^{obj_i}(s,a)$ values of the actions related to the elixir decrease as the daring factor decreases. In f gure 5, it is observed that the worst values are much lower than the limit introduced in (5). Therefore, although the $Q^{obj_i}(s,a)$ values of the actions related to the elixir are very high, according to (40) the values $Q_{fear}^{obj_i}(s,a)$ related to the elixir are going to vary as shown in the f gure.

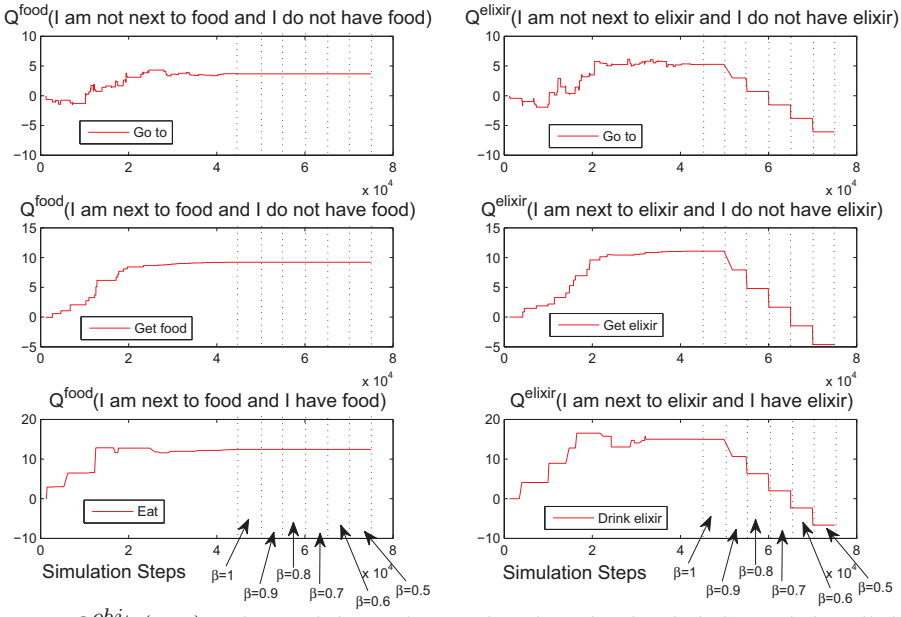

Figure 7: $Q_{fear}^{obj_i}(s,a)$ values of the actions related to the food (left) and the elixir (right), when the dominant motivation is the Hunger

These $Q_{fear}^{obj_i}(s,a)$ values imply that when the agent does not consider the possibility of being poisoned ($\beta = 1$) and it is hungry, and it is neither next to the food nor the elixir, the agent will go where the elixir is, take it and drink it. This is because the agent selects the action with the highest value in that state (f rst row of f gure 7) and this value corresponds to the action "go where the elixir is". Once the agent is in the new state (being next to the elixir), the action with the highest value is "get the elixir" (the other option would be "go where the food is"). Finally, from that state, to be next to the elixir and to have the elixir, the selected action is "drink the elixir". Again, the agent could select "go where the food is". It must be said that in every state the agent considers all the available actions related to every object. This f gure shows the values of the actions related to the food and the elixir because they are the highest ones but not because the agent has not the possibility to select any other action related to another object.

As the daring factor β varies, the agent changes its selection of actions. In the case of $\beta = 0.8$, when the agent is in the initial state, that is, not being next to the food nor to the elixir, the action with the highest $Q_{fear}^{obj_i}(s,a)$ value is "go where the food is". From that new state, being next to the food, the agent selects the action "get the food" and from there, the action with the highest value is "eat the food".

According to the values presented in f gure 7, when the agent is hungry and the daring factor is lower than 0.8, the agent will not select any action related to the

elixir.

Nevertheless, table 3 shows that the number of times that the agent drinks the elixir when $\beta = 0.8$ is 21. This is because the $Q_{fear}^{obj_i}(s,a)$ values analyzed are those obtained when the agent is hungry. In the case that the agent is thirsty, the $Q_{fear}^{obj_i}(s,a)$ values of water and elixir are very similar and may cause that, in some occasions, it keeps preferring the elixir.

On the other hand, different behaviours are obtained in this experiment for other inner states. When the agent has the Weakness as the dominant motivation, the $Q_{fear}^{obj_i}(s,a)$ values related to the elixir are now lower than the ones related to the medicine. As a result of this fact, the agent, when weak, always drinks medicine instead of elixir, no matter what the value of the daring factor is. Moreover, when the agent has no dominant motivation, the $Q^{obj_i}(s,a)$ values related to the rest of the objects, including the elixir, are lower than the ones related to the medicine. Therefore, when the agent is "ok", for all the values of the daring factor the agent prefers to drink medicine again.

In **summary**, in this experiment, when the agent does not consider the worst effect of the risky action, to be when the daring factor is $\beta = 1$, it prefers to drink the elixir when it is hungry or thirsty, since this action has a very high $Q^{obj_i}(s,a)$ value and, as a consequence, a high $Q_{fear}^{obj_i}(s,a)$ value. Nevertheless, when the agent is weak or when there is no dominant motivation, it prefers to drink medicine.

On the other hand, when the agent is risk-adverse, then the policy of behaviour changes. Depending on the value of the daring factor β, there will be a moment when the agent will not choose to drink the elixir again, since its $Q_{fear}^{obj_i}(s,a)$ value is very low. It is very interesting that the agent not only decides not to drink the elixir, but will not select any other action that leads to that situation. This is because, as shown in f gure 5, the values of "go for the elixir" and "take the elixir" are also lower than the limit L_m.

6 Conclusions

In this paper, the usefulness of the mechanism inspired by fear in order to learn to deal with risky actions is shown. The risk is related to negative rewards with a very low probability. In order to test this algorithm, it has been implemented on an autonomous agent. The agent lives in an environment where a dangerous object exists, the elixir, and the action of drinking it is considered risky according to our def nition.

The algorithm presented can be viewed as a risk-sensitive reinforcement learning algorithm. Using it, the agent tries to maximize a new Q_{fear} value that combines the optimal Q values given by Q-learning and the Q_{worst} values, when these last ones are lower than a certain limit. The daring factor β ponders the importance of the worst results and the optimal Q values. Therefore, this parameter def nes the risk-aversion of the agent.

As shown, the capacity of selecting the risk-aversion of the agent is the main difference with the worst-case criterion proposed by Heger [Heger, 1994]. In relation to other risk-sensitive approaches, such as the ones that use exponential utility functions [Howard and Matheson, 1972] [Coraluppi and Marcus, 1999] and expected value-minus-variance-criterion [Heger, 1994], we do not need the transition

probabilities, since we use a model-free approach. Moreover, we are also able to vary the daring factor during the life of the agent, since the learnt Q values do not depend on this factor. This fact makes the difference between our approach and the risk-sensitive reinforcement learning algorithm proposed by Mihatsch and Neuneier [Mihatsch and Neuneier, 2002].

Varying the value of the daring factor the results show that, when the agent is risk-seeking ($\beta \approx 1$), it learns to drink the elixir, despite its possible negative effects. Therefore, most of the time the agent is able to satisfy several drives at the same time. As a result, the average value of its wellbeing is very high, although there are some occasional drops due to poisoning. On the other hand, when the agent becomes risk-adverse (low values of β), it disregards the elixir and follows a "safe" policy of behaviour. This causes that the agent is not poisoned any more, with no drops in its wellbeing, improving its quality of life.

The mechanism proposed in this paper is essential for autonomous agents living in a complex environment, since some behaviours could compromise their integrity. Particularly, all superior animal has this kind of mechanism that helps them to avoid actions that could lead to death. In many cases, the learning process is associated to phylogenetics and therefore, it is not linked to the experiences of the individual animal. In our tests, the agent learns everything by trial and error. Nevertheless, it could be also possible to use the proposed mechanism to design agents initially programmed with the knowledge of other agents.

Acknowledgements

The authors gratefully acknowledge the funds provided by the Spanish Government through the project called "A New Approach to Social Robotics" (AROS), of MICINN (Ministry of Science and Innovation) and through the RoboCity2030-II-CM project (S2009/DPI-1559), funded by Programas de Actividades I+D en la Comunidad de Madrid and cofunded by Structural Funds of the EU.

References

[Cañamero, 2003] Cañamero, L. (2003). *Emotions in Humans and Artifacts*, chapter Designing emotions for activity selection in autonomous agents. MIT Press.

[Coraluppi and Marcus, 1999] Coraluppi, S. P. and Marcus, S. I. (1999). Risk-sensitive and minimax control of discrete-time, f nite-state markov decision processes. *Automatica*, 35:301–309.

[Frank et al., 2008] Frank, J., Mannor, S., and Precup, D. (2008). Reinforcement learning in the presence of rare events. In *The 25th International Conference on Machine Learning, Helsinki, Finland, 2008*.

[Gadanho, 1999] Gadanho, S. (1999). *Reinforcement Learning in Autonomous Robots: An Empirical Investigation of the Role of Emotions*. PhD thesis, University of Edinburgh.

[Gadanho and Custodio, 2002] Gadanho, S. and Custodio, L. (2002). Asynchronous learning by emotions and cognition. In *From Animals to Animats VII,*

Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior (SAB'02), Edinburgh, UK.

[Gautier and Boeree, 2005] Gautier, R. and Boeree, G. (2005). *Teorías de la personalidad: una selección de los mejores autores del S. XX*. Ed. UNIBE.

[Geibel, 2001] Geibel, P. (2001). Reinforcement learning with bounded risk. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML01)*, pages 162–169. Morgan Kaufmann Publishers, San Francisco,CA.

[Geibel and Wysotzki, 2005] Geibel, P. and Wysotzki, F. (2005). Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artif cial Intelligence Research*, 24:81–108.

[Heger, 1994] Heger, M. (1994). Consideration of risk in reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 105–111.

[Howard and Matheson, 1972] Howard, R. A. and Matheson, J. E. (1972). Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369.

[Kaelbling et al., 1996] Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artif cial Intelligence Research*, 4:237–285.

[Lorenz and Leyhausen, 1973] Lorenz, K. and Leyhausen, P. (1973). *Motivation of human and animal behaviour; an ethological view*, volume XIX. New York: Van Nostrand-Reinhold.

[Malfaz and Salichs, 2006] Malfaz, M. and Salichs, M. (2006). Learning behaviour-selection algorithms for autonomous social agents living in a role-playing game. In *Proccedings of theAISB'06: Adaptation in Artif cial and Biological Systems. University of Bristol, Bristol, England*.

[Malfaz and Salichs, 2009] Malfaz, M. and Salichs, M. (2009). Learning to deal with objects. In *Proccedings of the 8th International Conference on Development and Learning (ICDL 2009)*.

[Malfaz and Salichs, 2010] Malfaz, M. and Salichs, M. (2010). Using muds as an experimental platform for testing a decision making system for self-motivated autonomous agents. *Artif cial Intelligence and Simulation of Behaviour Journal (AISBJ)*, 2(1):21–44.

[Mihatsch and Neuneier, 2002] Mihatsch, O. and Neuneier, R. (2002). Risk-sensitive reinforcement learning. *Machine Learning*, 49(2-3):267 – 290.

[Salichs and Malfaz, 2011] Salichs, M. and Malfaz, M. (2011). A new approach to modeling emotions and their use on a decision making system for artif cial agents. *IEEE Transactions on Affective Computing*, In Press.

[Watkins, 1989] Watkins, C. J. (1989). *Models of Delayed Reinforcement Learning*. PhD thesis, Cambridge University, Cambridge, UK.

[Zimmerman, 2007] Zimmerman, B. (2007). http://www.coffeemud.org.

[Ávila García and Cañamero, 2004] Ávila García, O. and Cañamero, L. (2004). Using hormonal feedback to modulate action selection in a competitive scenario. In *Procceding of the 8th International Conference on Simulation of Adaptive Behavior (SAB'04)*.